



(12) **EUROPEAN PATENT APPLICATION**

(43) Date of publication:
18.10.2000 Bulletin 2000/42

(51) Int. Cl.⁷: **G06F 17/60**

(21) Application number: **00302844.6**

(22) Date of filing: **04.04.2000**

(84) Designated Contracting States:
**AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU
MC NL PT SE**
Designated Extension States:
AL LT LV MK RO SI

(72) Inventor: **Gawne, Stephen C.**
Waterloo, Ontario N2V 1P7 (CA)

(74) Representative:
Cleary, Fidelma et al
International IP Department
NCR Limited
206 Marylebone Road
London NW1 6LY (GB)

(30) Priority: **16.04.1999 US 292889**

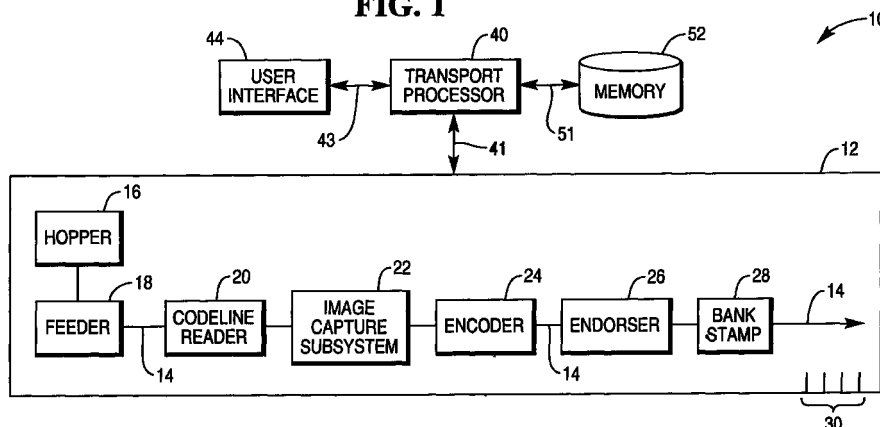
(71) Applicant:
NCR INTERNATIONAL INC.
Dayton, Ohio 45479 (US)

(54) **Method of operating a financial document processing system during exception recovery**

(57) A method of operating a document processing system having at least one executable transport application program, a document transport path along which documents can be transported, and a number of hardware devices disposed along the document transport path for processing documents moving along the document transport path in accordance with execution of the transport application program, comprises the steps of (a) processing a document at each hardware device as the document moves past the respective device, (b) at

each hardware device storing a respective status signal after the document has been processed by the particular hardware device, (c) suspending execution of the transport application program when an exception condition occurs, and (d) continuing execution of the transport application program to continue processing the document based upon the last stored status signal for the particular document.

FIG. 1



Description

[0001] The present invention relates to financial document processing systems, and is particularly directed to a method of operating a financial document processing system, such as an image-based check processing system, during exception recovery.

[0002] A typical image-based check processing system includes a check processing transport which has a document transport path and a number of different hardware devices positioned along the document transport path for performing specific document processing operations on documents moving downstream along the document transport path. The check processing system also includes a transport processor which executes a transport application program which is stored in memory to control operation of the devices positioned along the document transport path and thereby to control operation of the check processing transport.

[0003] From time to time, an exception condition may occur while processing documents on the check processing transport. Exception conditions include document jams, hardware failures, application errors, media low or empty, a pocket full, for example. Typically, when an exception condition occurs in the document transport path, any documents in the document transport path that have not been completely processed are manually located by an operator and removed from the document transport path. To avoid problems further downstream, the operator must ensure that all documents which have not been completely processed are removed from the document transport path. The removed documents must then be reprocessed in their original order. In known transport application programs, exception recovery programs are typically used to assist operators during exception recovery. In known exception recovery programs, much manual effort is still required of an operator to clear the exception condition. It would be desirable to provide an exception recovery program in which the manual effort required of an operator during exception recovery is reduced.

[0004] In accordance with one aspect of the present invention, a method of operating a document processing system having at least one executable transport application program, a document transport path along which documents can be transported, and a number of hardware devices disposed along the document transport path for processing documents moving along the document transport path in accordance with execution of the transport application program, comprises the steps of (a) processing a document at each hardware device as the document moves past the respective device, (b) at each hardware device storing a respective status signal after the document has been processed by the particular hardware device, (c) suspending execution of the transport application program when an exception condition occurs, and (d) continuing execu-

tion of the transport application program to continue processing the document based upon the last stored status signal for the particular document.

[0005] In accordance with another aspect of the present invention, a program storage medium readable by a computer having a memory is provided. The medium tangibly embodies one or more programs of instructions executable by the computer to perform method steps for operating a document processing system having at least one executable transport application program, a document transport path along which documents can be transported, and a number of hardware devices disposed along the document transport path for processing documents moving along the document transport path in accordance with execution of the transport application program. The method comprises the steps of (a) processing a document at each hardware device as the document moves past the respective device, (b) at each hardware device storing a respective status signal after the document has been processed by the particular hardware device, (c) suspending execution of the transport application program when an exception condition occurs, and (d) continuing execution of the transport application program to continue processing the document based upon the last stored status signal for the particular document.

[0006] An embodiment of the present invention will now be described by way of example, with reference to the accompanying drawings, in which:

Fig. 1 is a schematic block representation of an image-based check processing system embodying the present invention;

Fig. 2 is a schematic block representation of a portion of Fig. 1 and showing a memory in which software is stored;

Fig. 3 is a layered representation of software stored in the memory of Fig. 2;

Fig. 4 is a block representation of one layer of software shown in Fig. 3;

Fig. 5 is a representation of a transport configuration database used in the one layer of software of Fig. 4;

Figs. 6-8 are charts showing arrangement and relationships of certain data fields in a transport configuration database of Fig. 5;

Fig. 9 is a flowchart depicting processes carried out during configuring mode of the check processing system of Fig. 1;

Fig. 10 is a flowchart depicting processes carried out during operating mode of the check processing system of Fig. 1;

Fig. 11 is a flowchart depicting processes carried out by a zone verifier in the flowcharts of Figs. 9 and 10;

Figs. 12-16 are examples of zone verification performed in the flowchart of Fig. 11; and

Figs. 17 and 18 are flowcharts depicting processes

carried out during exception recovery in the check processing system of Fig. 1.

[0007] The present invention is directed to financial document processing systems, and is particularly directed to a method of operating a financial document processing system. The specific construction and use of the financial document processing system may vary. By way of example, a financial document processing system in the form of an image-based check processing system 10 is illustrated in Fig. 1. The check processing system 10 may be, for example, a sorting machine or a proof machine wherein financial documents such as checks are processed in a data processing operation.

[0008] As shown in Fig. 1, the check processing system 10 includes a check processing transport 12 having a document track which defines a document transport path 14 along which financial documents (i.e., items), such as checks, can be transported from an upstream end to a downstream end. The transport 12 includes a number of different hardware devices lying along the document transport path 14 for performing specific document processing operations on documents moving along the document transport path 14. The transport 12 includes a hopper 16 into which a stack of financial documents including checks are placed. A document feeder 18 adjacent the hopper 16 selectively feeds or drives each document from the stack of documents in the hopper to transport the document from the upstream end to the downstream end along the document transport path 14 to sorting bins 30 (i.e., pockets) located at the end of the document transport path.

[0009] The check processing system 10 further includes a codeline reader 20 such as a MICR reader located along the document transport path 14. The MICR reader 20 reads a MICR codeline from each check being processed in a known manner. Alternatively, the codeline reader may be an OCR reader instead of a MICR reader depending upon on the particular application. The check processing system 10 further includes an image capture subsystem 22 located along the document transport path 14. The image capture subsystem 22 captures an image of each document for a number of different purposes well known in the art. More specifically, the image capture subsystem 22 includes an imaging camera (not shown) which is controlled to capture images of documents moving along the document transport path 14.

[0010] An encoder 24 encodes missing fields on each check. An endorser 26 applies an endorsement in a known manner to each check. A bank stamp 28 stamps each check to identify the bank institution processing the check. The structure and operation of MICR readers, OCR readers, imaging cameras, encoders, endorsers, and bank stamps are well known and, therefore, will not be described.

[0011] Referring to Figs. 1 and 2, the check processing system 10 further includes a transport proc-

essor 40 and a user interface 44 which communicates via signals on line 43 (Fig. 1) with a microcomputer 42 of the transport processor 40. The user interface 44 includes a keyboard 46, a mouse 48, and a display 50, all of which communicate via signals on lines 43a, 43b, 43c (Fig. 2) with the microcomputer 42. The microcomputer 42 controls operation of the transport 12 via signals on line 41. Suitable microcomputers and memories are readily available in the marketplace. Their structure and operation are well known and, therefore, will not be described. The check processing system 10 also includes a memory 52 which communicates via signals on line 51 with the microcomputer 42. An operating system 54 is stored in the memory 52. Preferably, the operating system 54 comprises WINDOWS NT offered by Microsoft Corporation, located in Redmond, WA. A number of different transport application programs 56 are also stored in the memory 52. As shown in Fig. 2, there are "N" number of different transport application programs stored in the memory 52. Each of the transport application programs 56 is associated with a particular type of document processing work. For example, transport application #1 may be associated with one type of work, such as proof of deposit, while transport application #2 is associated with a different type of work, such as remittance processing. Another type of work may be sorting of items

[0012] A control program 60 is also stored in the memory 52. As best shown in Fig. 3, the control program 60 includes a layer of software that interfaces between the operating system 54 and a selected one of the transport application programs 56. The control program 60 allows the selected one of the transport application programs 56 to use the hardware devices of the transport 12. The control program 60 also allows different transports and/or different hardware devices to be used in the check processing system 10 shown in Fig. 1 without requiring changes to any of the transport application programs 56. Details and operation of the control program 60 are described hereinbelow.

[0013] Referring to Fig. 4, the control program 60 includes a number of interconnected components. The interconnected components include a configuration manager 64 which communicates with a graphical user interface of the operating system 54, a transport configuration database 66 which will be described in detail later, and a zone verifier 68 which will also be described in detail later. The configuration manager 64 allows a person, such as a system designer, to enter data into and view data from the transport configuration database 66 via the graphical user interface. The control program 60 also includes a transport controller 70 which processes data from the zone verifier 68 and data from the transport configuration database 66 to provide control data for the selected one of the transport application programs 56 to control the transport 12 and the different hardware devices which lie along the document transport path 14.

[0014] The control program 60 further includes an item control memory 72 which stores sequence numbers, MICR codelines, image data, encoder status, endorsement status, and bank stamp status associated with items which have been processed in accordance with the transport application program 56. The item control memory 72 also stores data relating to certain items during an exception recovery process. More specifically, the data stored in the item control memory 72 represents the current state of output operations (i.e., all completed and incomplete input/output operations) for each item involved in an exception condition along the document transport path 14. The control program 60 also includes an exception recovery program 74 which will be described in detail later.

[0015] Referring to Figs. 4 and 5, the transport configuration database 66 includes a transport device knowledge database (TDKB) and a number of different transport configuration sets of parameters. The TDKB contains zone information rules which will be described in detail later in connection with description of the zone verifier 68. The transport configuration sets of parameters contain device-related parameters for all types of devices which are contemplated to be used in the transport 12 of the check processing system 10. The transport configuration sets of parameters also include transport-related parameters for all types of transports which are contemplated to be used in the transport 12. The transport configuration sets of parameters further contain general document processing requirements for different types of documents which are expected to be processed by the transport 12.

[0016] During operation of the check processing system 10, one set of the transport configuration sets of parameters is selected to allow the selected one of the transport application programs 56 to control operation of the hardware device of the transport 12 in accordance with the selected set of parameters. The selected one of the different transport configuration sets of parameters will be referred to hereinafter as the "TCS".

[0017] As shown in Figs. 6-8, the TCS has a number of different data fields. The names of the data fields include "I/O category", "Property Page Number", "Data Format Requirement(s)", "Field Definition(s)", and "Device Specific Linkage(s)". Each I/O category corresponds to a specific document processing function to be performed by a hardware device of the transport 12. The I/O categories may include a codeline reading function of the transport 12 such as shown in Fig. 6, an imaging function of the transport such as shown in Fig. 7, and an encoding function of the transport such as shown in Fig. 8.

[0018] Referring to Fig. 6, a number of different property pages (e.g., "J" number of property pages as shown in Fig. 6) is associated with the codeline reading function of the transport 12. For each property page, a codeline data format requirement(s) field, a field definition(s) field, and a device specific linkage(s) field are

associated with that particular property page. The codeline data format requirement(s) field may include subfields which define (i) codeline direction, and (ii) use spaces, for examples. The field definition(s) field may include subfields which define X/Y zone coordinates which indicate location on a document where reading is to be performed, for example. The information contained in the subfields of the codeline data format requirement(s) field and the information contained in the subfields of the field definition(s) field are well understood by one skilled in the art and, therefore, will not be described.

[0019] The device specific linkage(s) field includes subfields which define transport model numbers, such as 7780 and 7790, both available from NCR Corporation, located in Dayton, Ohio. For each transport model number, there may be a number of different types of MICR readers associated with that particular transport model number. As shown in Fig. 6, an E13B dual MICR reader is associated with transport model number 7780. One specific characteristic is associated with the E13B dual MICR reader which is associated with transport model number 7780. The specific characteristic is enable E13 reject/repair optional assist which is a characteristic well understood by those skilled in the art and, therefore, will not be described.

[0020] Also, as shown in Fig. 6, an E13B MICR reader is associated with transport model number 7790. Three specific characteristics are associated with the E13B MICR reader which is associated with transport model number 7790. The three specific characteristics are (i) stop feeding on black band detect, (ii) MICR repair type, and (iii) read option. Similarly, a CMC7 reader is also associated with transport model number 7790. Two specific characteristics are associated with the CMC7 MICR reader which is associated with transport model number 7790. The two specific characteristics are (i) font switch string, and (ii) font switch position. These characteristics of E13B MICR readers and CMC7 MICR readers as well as other characteristics of MICR readers are well understood by those skilled in the art and, therefore, will not be described.

[0021] Although only two transport model numbers are shown in the device specific linkage(s) field in Fig. 6, it should be understood that any number of transport models having any particular type of MICR reader could be included in the device specific linkage(s) field. This will depend upon what combinations of different transport model numbers and different MICR reading devices are expected to be used in the check processing system 10.

[0022] Referring to Fig. 7, a number of different property pages (e.g., "K" number of property pages shown in Fig. 7) is associated with the imaging function of the transport 12. For each property page, an image data format requirement(s) field, a field definition(s) field, and a device specific linkage(s) field are associated with that particular property page. The image data

format requirement(s) field may include subfields which define (i) TIFF (Tagged Image File Format by Adobe Systems Incorporated, located in San Jose, California) tags used in an image header, (ii) image orientation, (iii) image color such as black/white, gray, or color. The field definition(s) field may include subfields which define X/Y zone coordinates which indicate location on a document where imaging is to be performed, for example. The information contained in the subfields of image data format requirement(s) and the information contained in the subfields of field definition(s) are well understood by one skilled in the art and, therefore, will not be described.

[0023] The device specific linkage(s) field includes subfields which define transport model numbers, such as 7780, 7790, and 7731, all available from NCR Corporation, located in Dayton, Ohio. For each transport model number, there may be a number of different types of image capture devices associated with that particular transport model number. As shown in Fig. 7, an upstream camera is associated with transport model number 7780. Two specific characteristics are associated with the upstream camera which is associated with transport model number 7780. The two specific characteristics are (i) enable binary cropping, and (ii) gray compression quality factor. These two characteristics of imaging cameras as well as other characteristics of imaging cameras are well understood to those skilled in the art and, therefore, will not be described.

[0024] Also, as shown in Fig. 7, a bi-level image module is associated with transport model number 7790. Two specific characteristics are associated with the upstream camera which is associated with transport model number 7780. The two specific characteristics are (i) front bi-level threshold, and (ii) rear bi-level threshold. Similarly, an image lift module is associated with transport model number 7731. Two specific characteristics are associated with the upstream camera which is associated with transport model number 7731. The two specific characteristics are (i) enable front gray cropping, and (ii) rear gray quality compression factor. The specific characteristics of imaging module as well as other characteristics of imaging modules are well understood to those skilled in the art and, therefore, will not be described.

[0025] Although only three transport model numbers are shown in the device specific linkage(s) field in Fig. 7, it should be understood that any number of transport models having any particular type of imaging camera/module could be included in the device specific linkage(s) field. This will depend upon what combinations of different transport model numbers and different imaging cameras/modules are expected to be used in the check processing system 10.

[0026] Referring to Fig. 8, a number of different property pages (e.g., "L" number of property pages shown in Fig. 8) is associated with the encoding function of the transport 12. For each property page, an

encode data format requirement(s) field, a field definition(s) field, and a device specific linkage(s) field are associated with that particular property page. The encode data format requirement(s) field may include subfields which define (i) encode fonts, and (ii) default encode data. The field definition(s) field may include subfields which define X/Y zone coordinates which indicate location on a document where encoding is to be performed, for example. The information contained in the subfields of encode data format requirement(s) and the information contained in the subfields of field definition(s) are well understood by one skilled in the art and, therefore, will not be described.

[0027] The device specific linkage(s) field includes subfields which define transport model numbers, such as 7780 and 7790, both available from NCR Corporation, located in Dayton, Ohio. For each transport model number, there may be a number of different types of encoding devices associated with that particular transport model number. As shown in Fig. 8, a high speed encoder is associated with transport model number 7780. Two specific characteristics are associated with the high speed encoder which is associated with transport model number 7780. The two specific characteristics are (i) enable skew detection, and (ii) enable dog ear detection. These two characteristics of encoding devices as well as other characteristics of encoding devices are well understood to those skilled in the art and, therefore, will not be described.

[0028] Also, as shown in Fig. 8, a high speed encoder is associated with transport model number 7790. Two specific characteristics are associated with the high speed encoder which is associated with transport model number 7790. The two specific characteristics are (i) enable dog ear detection, and (ii) outsort on encode error. The specific characteristics of encoding devices as well as other characteristics of encoding devices are well understood to those skilled in the art and, therefore, will not be described.

[0029] Although only two transport model numbers are shown in the device specific linkage(s) field in Fig. 8, it should be understood that any number of transport models having any particular type of encoding devices could be included in the device specific linkage(s) field. This will depend upon what combinations of different transport model numbers and different encoding devices are expected to be used in the check processing system 10.

[0030] Also, although only three I/O categories (codeline reading, imaging, and encoding) are described hereinabove, it should be understood that other I/O categories, such as document feeding, endorsing, and bank stamping, may be included as well.

[0031] Referring to Fig. 9, a flowchart depicts operation of the check processing system 10 in configuring mode when a system designer is inputting data into the transport configuration sets of parameters in the transport configuration database 66 to define parameters for

different transports and/or hardware devices which may be used in the check processing system 10.

[0032] In step 100, all transport configuration sets of parameters stored in the transport configuration database 66 are displayed on the display 50 for the system designer to view. In step 102, the system designer selects one of the transport configuration sets of parameters (i.e., the TCS) depending upon the particular document processing work to be performed. The program then proceeds to step 104 in which for each property page within each I/O category, a representative image of the physical item to be processed is displayed on the display 50 for the system designer to view. The system designer then draws zones on the representative image of the physical item to be processed as shown in step 106. The system designer also selects data format requirements and specific hardware device settings as shown in steps 108 and 110.

[0033] The zone verifier 68 is then invoked in step 112. Briefly, when the zone verifier 68 is invoked in step 112, a representative image of how the physical item will be processed on the transport 12 will be displayed on the display 50 for the system designer to view. At this point, the system designer makes any needed adjustments to the data format requirements and/or specific hardware device settings. Further details of operation of the zone verifier 68 will be described later. The system designer continues the above-described steps for each physical item type within each I/O category until the configuring process is completed.

[0034] Referring to Figs. 10A, 10B, and 10C, a flow-chart depicts operation of the check processing system 10 in operating mode when the transport 12 is operating in accordance with the selected one of the transport application programs 56 stored in the memory 52. In step 200, all of the transport application programs 56 which can be run are displayed on the display 50 for an operator to view. The operator then selects one of the transport application programs 56 based upon the particular document processing work which is to be performed. The process then proceeds to step 206 in which the selected transport application program selects the TCS for that particular transport application program. Then, in step 208, the transport controller 70 determines what hardware devices are actually on the transport 12. The process then proceeds to step 210 in which for each field definition within each property page within each I/O category, the transport controller 70 determines if there is a match between a device specific linkage and a hardware device which is actually being used on the transport 12. More specifically, the transport controller 70 determines if there is a match by determining if there is at least one hardware device on the transport 12 capable of performing function of the I/O category defined by the property page.

[0035] The process then proceeds to step 212 in which a determination is made as to whether a match occurred in step 210. If the determination in step 212 is

negative, the process proceeds directly to step 218. However, if a match is found as determined in step 212, the process proceeds to step 214 in which the transport controller 70 invokes the zone verifier 68 to make any needed adjustments to settings of the particular property page under consideration. These adjustments are made based upon zone information rules stored in the TDKB to be described in detail later. As shown in step 216, any adjustments to settings of the particular property page are stored in the transport configuration database 66.

[0036] The process then proceeds to step 218 in which a determination is made in step 218 as to whether there are more field definitions to consider. Also, a determination is made in step 220 as to whether there are more property pages to consider, and a determination is made in step 222 as to whether there are more I/O categories to consider. When all field definitions, all property pages, and all I/O categories have been considered, the process proceeds to step 224. In step 224, a determination is made as to whether there were any settings of the TCS which could not be processed. If the determination in step 224 is affirmative, the process proceeds to step 228 to communicate to the operator that there is an error present. At this point, the operator terminates the process. Otherwise, if the determination in step 224 is negative, the process proceeds to step 226.

[0037] In step 226, the selected transport application program selects the applicable property page from each applicable I/O category and is now ready to process items. The process then proceeds to step 232 in which for each I/O category and the corresponding property page which was selected in step 226 the transport controller 70 obtains from memory (i.e., the transport configuration database 66) the stored options for the selected property page. This data is then sent to the hardware device on the transport 12 to control that device accordingly.

[0038] The process then proceeds to step 234 to determine if there are more I/O categories to consider. If the determination is affirmative, then the process loops back to step 230 to consider the next I/O category. If the determination in step 234 is negative (indicating that all I/O categories have been considered), then the process proceeds to step 236 in which the selected transport application program requests an item to be fed from the hopper 16. The transport controller 70 indicates when all inputs are complete for that item, as shown in Fig. 238. After all inputs are completed for the item, the selected transport application program provides outputs to the device and the associated pocket to control that device and that pocket accordingly, as shown in step 240. A determination is then made in step 244 as to whether there are more items to process. If the determination in step 244 is affirmative, then the process loops back to step 236 to process the next item. Otherwise, all items have been processed and the process

ends.

[0039] A number of advantages are provided when either configuring or operating the financial document processing system 10 as described hereinabove. One advantage is that development of the transport application programs 56 stored in the memory 52 is simplified. Transport application program development is simplified because changes to the transport application programs 56 are in most instances not needed when either the transport 12 and/or a hardware device is changed, added, or removed. Accordingly, another advantage is that custom programming of the transport application programs 56 is not needed to accommodate different types of transports and/or different types of hardware devices used in the transports. The transport processor 40 is able to control operation of the transport and the hardware devices in the transport 12 independent of the type of transport and/or type(s) of device(s) used in the transport.

[0040] Another advantage is that hardware devices in the transport are allocated, without human intervention, for a document processing job to be performed. The hardware devices in the transport are allocated based upon the type of work to be done (i.e., type of document to be processed) and the types of the actual devices used in the transport. For example, if a document processing job to be performed requires three lines of endorsement data to be printed and only three single-line endorsers are available on the transport, then the endorsement data could be allocated to the three endorsers such that each endorser prints one of the three required lines of endorsement data.

[0041] Still another advantage is that a single transport configuration database 66 is provided to support all types of transports and all types of hardware devices contemplated to be used in the check processing system 10. Since only a single configuration database 66 is used to define all types of transports and all types of hardware devices, maintenance and updating of data associated with multiple hardware configurations are more easily accomplished.

[0042] Referring to Figs. 11A and 11B, a flowchart depicts operation of the zone verifier 68 shown in Fig. 4 in either configuring mode or operating mode of the check processing system 10. In step 300, the zone verifier 68 receives a hardware device model number from the transport controller 70 if in operating mode or from a system designer via the configuration manager 64 if in configuring mode. The process proceeds to step 302 in which zone information rules are obtained from the TDKB of the transport configuration database 66 based upon the hardware device model number received from either the transport controller 70 or the system designer. A determination is made in step 304 as to whether the check processing system 10 is in configuring mode or in operating mode.

[0043] If the determination in step 304 is such that the system 10 is in configuring mode, the process pro-

ceeds to step 306 in which an image of the item is received and then displayed on the display 50 as shown in step 308. The process then proceeds to step 310 in which inputs are received from a system designer via the mouse 48. The inputs from the system designer via the mouse 48 identifies zone coordinates of either a designer selected or a designer created zone on the image of the item displayed on the display 50. The process then proceeds to step 312. However, if the determination in step 304 is such that the system 10 is in operating mode, then the process proceeds to step 314 in which zone coordinates are received from the associated property page in the TCS before proceeding to step 312.

[0044] In step 312, the zone coordinates received from either the mouse 48 when in configuring mode or the property page in the TCS when in operating mode are compared against the zone information rules obtained from the TDKB of the transport configuration database 66. A determination is made in step 316 as to whether the zone coordinates satisfy the zone information rules based upon the comparisons made in step 312. If the determination in step 316 is such that the zone coordinates do satisfy the zone information rules and the system 10 is in operating mode, then the process is completed. If the determination in step 316 is such that the zone coordinates do satisfy the zone information rules and the system 10 is in configuring mode, then the zone coordinates are displayed on the image of the item displayed on the display 50 to allow the system designer to view before the process is completed.

[0045] However, if the determination in step 316 is such that the zone coordinates do not satisfy the zone information rules and the system 10 is in either configuring mode or operating mode, then the process proceeds to step 320. In step 320, the zone coordinates are changed to satisfy the zone information rules. Five examples of how the zone coordinates may be changed to satisfy certain zone information rules will be discussed hereinbelow with reference to Figs. 12-16. The process then proceeds to step 322 in which the changed zone coordinates are sent to the transport application program, if in operating mode, via the transport controller 70 for use in processing items. The changed zone coordinates are displayed on the image of the item displayed on the display 50, if in configuring mode, to allow the system designer to view the changed zone coordinates as shown in step 324.

[0046] Referring to Figs. 12-16, a number of different examples of zone verification are illustrated to show how the zone verifier 68 shown in Fig. 4 works using zone information rules obtained from the TDKB.

[0047] Figs. 12A and 12B show how zone coordinates are changed using zone information rules which are directed to an area of interest (AOI). The AOI has coordinates of (54, 35), (10, 35), (54, 18), and (10, 18), as shown in Figs. 12A and 12B. The zone information rules are (i) all zones should be inside the AOI, (ii) min-

imum zone height is 5, (iii) maximum zone height is 5, and (iv) zone left edge should be on grid which has a grid width of 3. The original zone coordinates are (50, 40), (20, 40), (50, 30), and (20, 30) as shown in Fig. 12A. After the zone information rules are applied, the changed zone coordinates become (51, 35), (18, 35), (51, 30), and (18, 30). It should be noted that the changed zone coordinates are now within the AOI, have a zone height of 5, and a left edge at 51. The left edge is at 51 because of the rules which require the grid width to be 3 and the left edge to be on the grid.

[0048] Figs. 13A and 13B show how zone coordinates are changed using zone information rules which are directed to zone overlap of two zones (ZONE1 and ZONE2). Original position of ZONE1 and original position of ZONE2 are shown in Fig. 13A. Original position of ZONE1 has coordinates of (50, 40), (20, 40), (50, 30), and (20, 30), and original position of ZONE2 has coordinates of (46, 36), (10, 36), (46, 26), and (10, 26), as shown in Fig. 13A. The zone information rules are (i) all zones should not overlap each other, and (ii) left edges of all zones should be the same. After the zone information rules are applied, the changed zone coordinates of ZONE1 become (50, 40), (20, 40), (50, 30), and (20, 30) and the changed zone coordinates of ZONE2 become (50, 30), (14, 30), (50, 20), and (14, 20), as shown in Fig. 13B. It should be noted that after the zone coordinates of ZONE1 and ZONE2 do not overlap and are aligned at their left edges after their zone coordinates are changed. Figs. 14A and 14B show how zone coordinates are changed using zone information rules which are directed to providing a minimum gap between the left edges of two zones (ZONE1 and ZONE2) and the right edges of the two zones. Original position of ZONE1 and original position of ZONE2 are shown in Fig. 14A. Original position of ZONE1 has coordinates of (50, 40), (20, 40), (50, 30), and (20, 30), and original position of ZONE2 has coordinates of (46, 24), (20, 24), (46, 14), and (20, 14), as shown in Fig. 14A. The zone information rule is that the minimum gap between any left edges and any right edges is 8. After the zone information rule is applied, the changed zone coordinates of ZONE1 remain the same and the zone coordinates of ZONE2 are changed to (42, 24), (12, 24), (42, 14), and (12, 14), as shown in Fig. 14B. It should be noted that there is a minimum gap of 8 between the left edge of ZONE1 and the left edge of ZONE2 and that there is a minimum gap of 8 between the right edge of ZONE1 and the right edge of ZONE2 after the zone coordinates of ZONE1 and ZONE2 are changed. It should also be noted that the gap between the left edge of ZONE2 and the right edge of ZONE1 is greater than the minimum allowable value of 8.

[0049] Figs. 15A and 15B show how zone coordinates are changed using zone information rules which are directed to at least two zones and an area of interest (AOI). The AOI has coordinates of (54, 34), (10, 34), (54, 18), and (10, 18), as shown in Figs. 15A and 15B.

Original position of ZONE1 and original position of ZONE2 are shown in Fig. 15A. ZONE1 is partially within the AOI and ZONE2 is entirely outside of the AOI. Original position of ZONE1 has zone coordinates of (50, 40), (20, 40), (50, 30), and (20, 30), as shown in Fig. 15A. The zone information rules are (i) verify only one zone, and (ii) all zones should be inside AOI. After the zone information rules are applied, the zone coordinates of ZONE1 are changed to (50, 34), (20, 34), (50, 24), and (20, 24). It should be noted that ZONE1 now lies entirely within the AOI. The zone coordinates of ZONE2 are not changed since the maximum number of zones to be considered is one as per the zone information rules. ZONE2 still lies entirely outside of the AOI.

[0050] Figs. 16A and 16B show how zone coordinates are changed using zone information rules which are directed to selecting one of two areas of interest (AOI 1 and AOI 2) as shown in Figs. 16A and 16B. The zone information rules are (i) select AOI 2, and (ii) all zones should be inside AOI 2. The zone coordinates of AOI 2 are (30, 12), (2, 12), (30, 2), and (2, 2) as shown in Figs. 16A and 16B. After the zone information rules are applied, the zone coordinates of the zone are changed to (30, 12), (10, 12), (30, 8), and (10, 8). It should be noted that the zone now lies within AOI 2.

[0051] A number of advantages are provided when data zones are drawn, verified, and adjusted if needed in accordance with the processes described hereinabove. One advantage is that guesswork is taken out of the task of setting up data zones for a particular transport application. Another advantage is that since operation of the zone verifier 68 is based upon a set of rules contained in a database, new hardware devices for use in the transport can be supported by simply adding new rules to the set of rules contained in the database without requiring any changes to the zone verifier 68 itself. Still another advantage is that the process of setting up data zones for a particular transport application is less prone to operator error and requires less time to accomplish.

[0052] A number of advantages are also provided when the zone verifier 68 is used in operating mode. One advantage is that zone coordinates which are sent to the hardware devices from the transport application program via the transport controller 70, are acceptable to the hardware devices. This eliminates disruptions that may happen if incorrect zone coordinates are sent to the hardware devices.

[0053] Referring to Figs. 17 and 18, a flowchart depicts operation of the exception recovery program 74 shown in Fig. 4 during an exception recovery condition in the check processing system 10.

[0054] When an exception condition occurs along the document transport path 14, an exception recovery process is initiated. Fig. 17 is a flowchart which depicts typical normal operation of the check processing system of Fig. 1. Fig. 18 is a flowchart which depicts the exception recovery program 74 which is invoked by the

program of Fig. 17 when an exception condition occurs.

[0055] In step 400, a sequence number is assigned to each item which is processed along the document transport path 14. The MICR codeline of the item is then read in accordance with the corresponding property page as already described hereinabove, as shown in step 402. A determination is then made in step 404 as to whether an exception condition has occurred after reading the MICR codeline for the particular item. If the determination in step 404 is negative, the process proceeds directly to step 406. Otherwise, if a determination is made in step 404 that an exception condition such as an exception condition has occurred, then the process proceeds to step 408 to invoke the exception recovery program of Fig. 18 which will be described later.

[0056] In step 406, an image of the item is lifted in accordance with the corresponding property page as already described hereinabove. A determination is then made in step 410 as to whether an exception condition has occurred after lifting the image for the particular item. If the determination in step 410 is negative, the process proceeds directly to step 412. Otherwise, if a determination is made in step 410 that an exception condition has occurred, then the process proceeds to step 414 to invoke the exception recovery program of Fig. 18.

[0057] In step 412, the item is encoded in accordance with the corresponding property page as already described hereinabove. A determination is then made in step 416 as to whether an exception condition has occurred after encoding the particular item. If the determination in step 416 is negative, the process proceeds directly to step 418. Otherwise, if a determination is made in step 416 that an exception condition has occurred, then the process proceeds to step 420 to invoke the exception recovery program of Fig. 18.

[0058] In step 418, the item is endorsed in accordance with the corresponding property page as already described hereinabove. A determination is then made in step 422 as to whether an exception condition has occurred after endorsing the particular item. If the determination in step 422 is negative, the process proceeds directly to step 424. Otherwise, if a determination is made in step 422 that an exception condition has occurred, then the process proceeds to step 426 to invoke the exception recovery program of Fig. 18.

[0059] In step 424, the item is pocketed in accordance with the corresponding property page as already described hereinabove. A determination is then made in step 428 as to whether an exception condition has occurred after pocketing the particular item. If the determination in step 428 is negative, the process proceeds directly to step 430. Otherwise, if a determination is made in step 428 that an exception condition has occurred, then the process proceeds to step 432 to invoke the exception recovery program of Fig. 18.

[0060] In step 430, a determination is made as to whether there are any more items to process. If the

determination is affirmative, then the process loops back to the start to process the next item to be processed. However, if the determination in step 430 is negative, then all items have been processed and the process terminates.

[0061] The flowchart of Fig. 18 depicts operation of the exception recovery program 74 referred to hereinabove. In step 500, the operator clears the document transport path 14 by removing all items from the document transport path. Then, in step 502, the transport controller 70 determines all items which need to be recovered. This determination is made based upon data contained in the item control memory 72 as is known. All items which need to be recovered are displayed in order of their sequence numbers as shown in step 504.

[0062] For each item which needs to be recovered, the item is hand dropped into the document transport path 14, as shown in step 506. The process then proceeds to step 508. In step 508, all required data as defined by the property pages which correspond to an expected item to be recovered from the document transport path 14 is obtained by the transport controller 70. The process then proceeds to step 510 in which a determination is made as to whether the data obtained in step 508 from the hand dropped item and the data associated with the expected item match. If the determination in step 510 is negative, then the process proceeds to step 512 in which an error message is displayed for the operator to view before looping back to step 506. At this point, the operator needs to manually obtain the correct item to hand drop into the document transport path 14. Otherwise, the process cannot proceed any further.

[0063] However, if the determination in step 510 is affirmative, then the process proceeds to step 514. In step 514, the transport controller 70 determines the last completed I/O device operation which was performed for that particular hand dropped item. The transport controller 70 makes this determination using data from the item control memory 72 based upon the sequence numbers of items. As previously described, data contained in the item control memory 72 is based upon data received from the transport 12 including the hardware devices used in the transport. More specifically, after an item is processed by a particular hardware device, a respective status signal is stored in the item control memory 72 to indicate that the item has been completely processed by that particular hardware device. It should be apparent that if an exception condition occurs, then the most recent status signal which was stored for the item is indicative of the corresponding hardware device performing the last I/O operation on the item. After the determination is made in step 514, the process proceeds to step 516.

[0064] In step 516, the transport controller 70 continues processing of the particular hand dropped item from the hardware device which immediately follows the hardware device which performed the last completed

I/O operation on that item as determined in previous step 514. If there are any more items to be considered during exception recovery, as shown in step 518, then the process terminates relative to that particular dropped item and loops back to step 506 in which the next item that needs to be recovered is hand dropped into the document transport path 14. Otherwise, if there are no more items to be considered during exception recovery, the process terminates and returns back to the process shown in the flowchart of Fig. 17.

[0065] A number of advantages are provided when the above-described process is carried out during exception recovery. One advantage is that an operator need not be concerned with settings of any of the hardware devices when item is to be hand dropped into the document transport path 14 during exception recovery. The operator need not be concerned with settings of any of the hardware devices because the transport controller 70 already knows which hardware device performed the last completed I/O operation on the particular item to be hand dropped, as determined in step 514. Accordingly, the transport controller 70 knows that any hardware device preceding the hardware device which performed the last completed I/O operation need not be operated at all to process the item to be hand dropped. The transport controller 70 also knows that only hardware devices (if any) following the hardware device which performed the last completed I/O operation will be operated to process the item to be hand dropped. Since an operator need not be concerned with settings of any of the hardware devices when an item is to be hand dropped into the document transport path 14, less manual effort is required of the operator during the exception recovery process. This results in savings of time and labor and, therefore, costs associated with operating the check processing system 10.

[0066] It is contemplated that any combination of the above-described programs be available on portable storage media, such as a compact disc read only memory (CDROM). The programs on a CDROM may be installed on different financial document processing systems to provide these systems with corresponding capabilities as described hereinabove. It is also contemplated that the above-described programs may be used in other types of financial document processing systems, such as non-image-based systems, for example.

Claims

1. A method of operating a document processing system having at least one executable transport application program, a document transport path along which documents can be transported, and a number of hardware devices disposed along the document transport path for processing documents moving along the document transport path in accordance with execution of the transport applica-

tion program, the method characterized by the steps of:

- (a) processing a document at each hardware device as the document moves past the respective device;
 - (b) at each hardware device storing a respective status signal after the document has been processed by the particular hardware device;
 - (c) suspending execution of the transport application program when an exception condition occurs; and
 - (d) continuing execution of the transport application program to continue processing the document based upon the last stored status signal for the particular document.
2. A program storage medium readable by a computer having a memory, the medium tangibly embodying one or more programs of instructions executable by the computer to perform method steps for operating a document processing system having at least one executable transport application program, a document transport path along which documents can be transported, and a number of hardware devices disposed along the document transport path for processing documents moving along the document transport path in accordance with execution of the transport application program, the method characterized by the steps of:
 - (a) processing a document at each hardware device as the document moves past the respective device;
 - (b) at each hardware device storing a respective status signal after the document has been processed by the particular hardware device;
 - (c) suspending execution of the transport application program when an exception condition occurs; and
 - (d) continuing execution of the transport application program to continue processing the document based upon the last stored status signal for the particular document.

FIG. 1

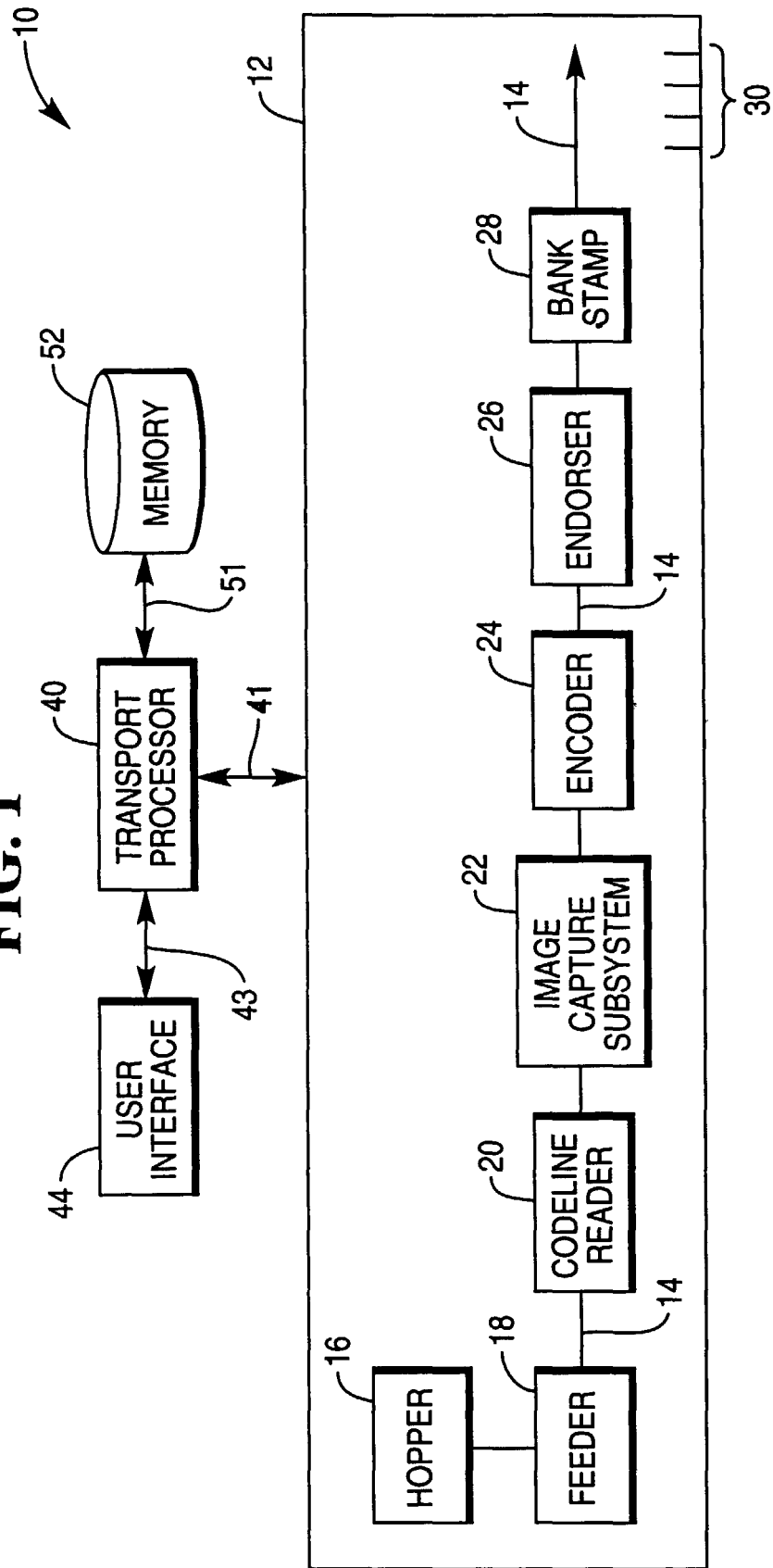


FIG. 2

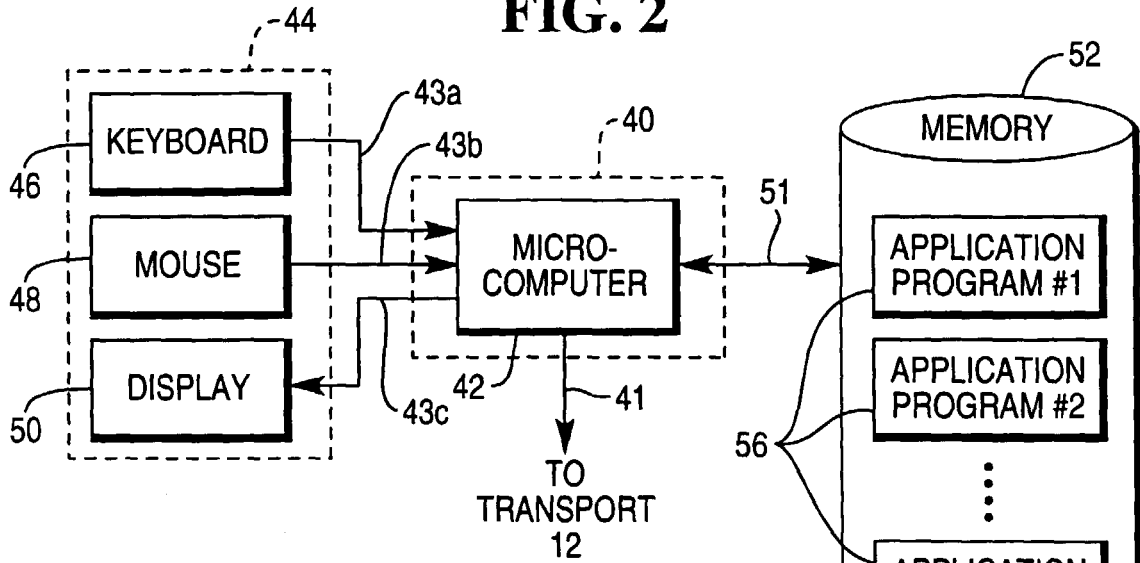


FIG. 3

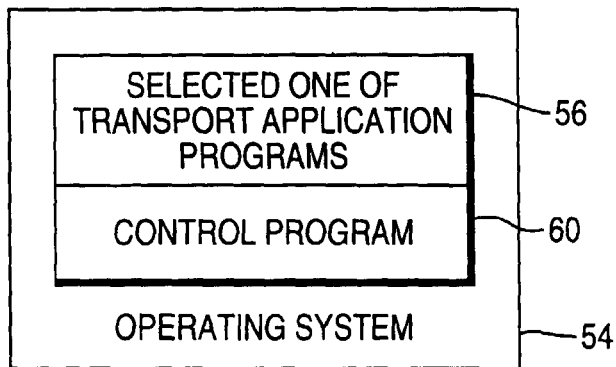


FIG. 5

TRANSPORT CONFIGURATION DATABASE

- TRANSPORT DEVICE KNOWLEDGE BASE
- TRANSPORT CONFIGURATION SETS OF PARAMETERS

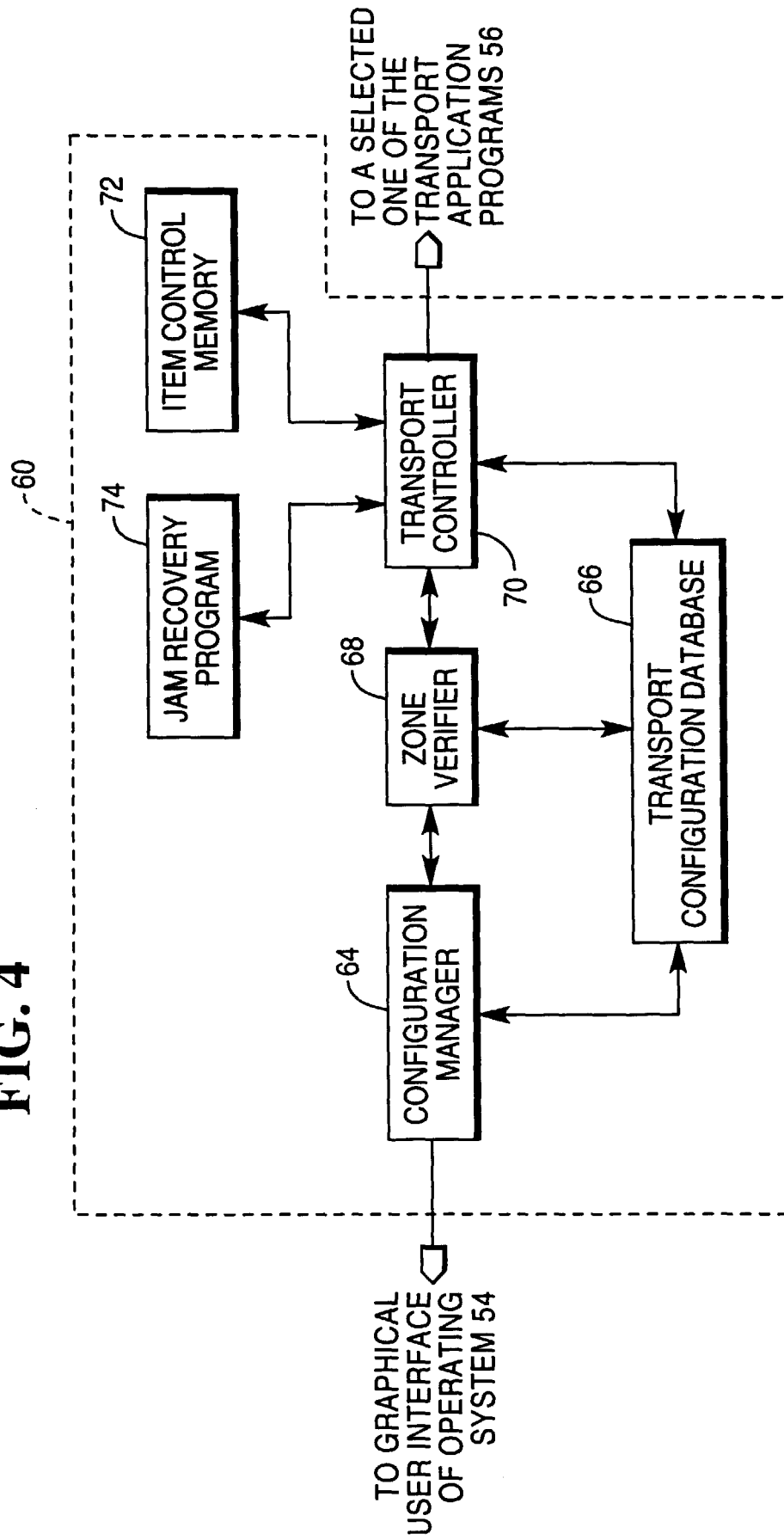
FIG. 4

FIG. 6**TRANSPORT CONFIGURATION SET (TCS)**

<u>I/O CATEGORY</u>	<u>PROPERTY PAGE #</u>	<u>CODELINE DATA FORMAT REQUIREMENT(S)</u>	<u>FIELD DEFINITION(S)</u>	<u>DEVICE SPECIFIC LINKAGE(S)</u>
CODELINE READING	1	• CODELINE DIRECTION	• XY ZONE	• 7780
	2	• USE SPACES? Y/N	COORDINATES	• E13B DUAL MICR
	.			•• ENABLE E13B REJECT/ REPAIR OPTIONAL ASSIST
	.			
	J			• 7790
				•• E13B MICR
				••• STOP FEEDING ON BLACK BAND DETECT
				••• MICR REPAIR TYPE <FORMAT & PLUG>
				••• READ OPTION ANALOG/DIGITAL
				•• CMC7 MICR
				••• FONT SWITCH STRING
				••• FONT SWITCH POSITION

FIG. 7

TRANSPORT CONFIGURATION SET (TCS)

I/O CATEGORY	PROPERTY PAGE #	IMAGE DATA FORMAT REQUIREMENT(S)	FIELD DEFINITION(S)	DEVICE SPECIFIC LINKAGE(S)
IMAGING	1	• TIFF TAGS USED IN IMAGE HEADER	• XY ZONE COORDINATES	• 7780
	2	• IMAGE ORIENTATION		•• UPSTREAM CAMERA
	.	• IMAGE COLOR		••• ENABLE BINARY CROPPING
	.	•• BLACK & WHITE		••• GRAY COMPRESSION QUALITY FACTOR
	.	•• GRAY		
	K	•• COLOR		
				• 7790
				•• BI-LEVEL IMAGE MODULE
				••• FRONT BI-LEVEL THRESHOLD
				••• REAR BI-LEVEL THRESHOLD
				• 7731
				•• IMAGE LIFT MODULE
				••• ENABLE FRONT GRAY CROPPING
				••• REAR GRAY QUALITY COMPRESSION FACTOR

FIG. 8

TRANSPORT CONFIGURATION SET (TCS)				
I/O CATEGORY	PROPERTY PAGE #	ENCODE DATA FORMAT REQUIREMENT(S)	FIELD DEFINITION(S)	DEVICE SPECIFIC LINKAGE(S)
ENCODING	1	• ENCODE FONT	• X/Y ZONE	• 7780
	2	• DEFAULT ENCODE DATA	• COORDINATES	• HIGH SPEED ENCODER
	.			•• ENABLE SKEW DETECTION
	.			••• ENABLE DOG EAR DETECTION
	L			
				• 7790
				•• HIGH SPEED ENCODER
				••• ENABLE DOG EAR DETECTION
				••• OUTSORT ON ENCODE ERROR

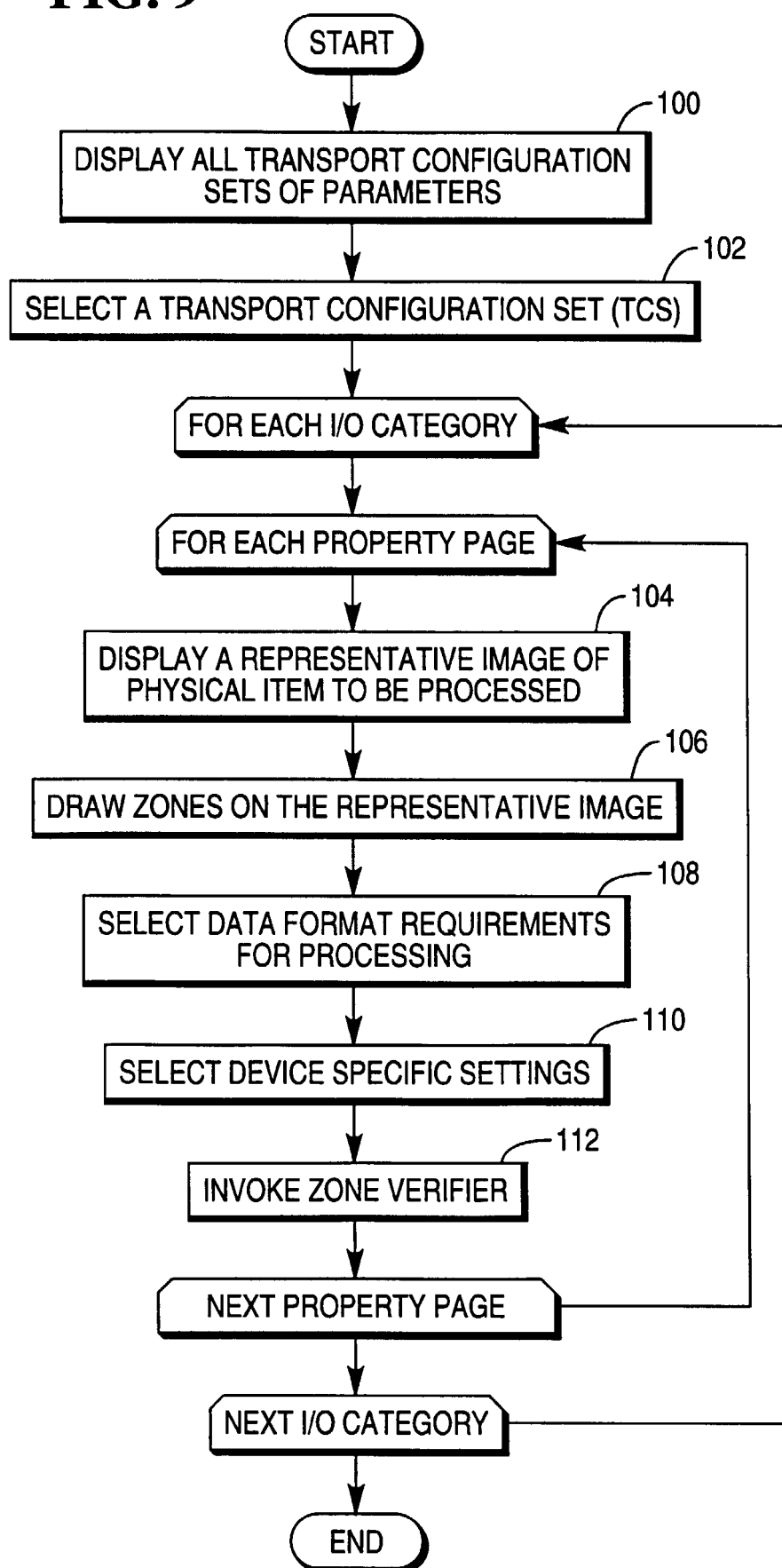
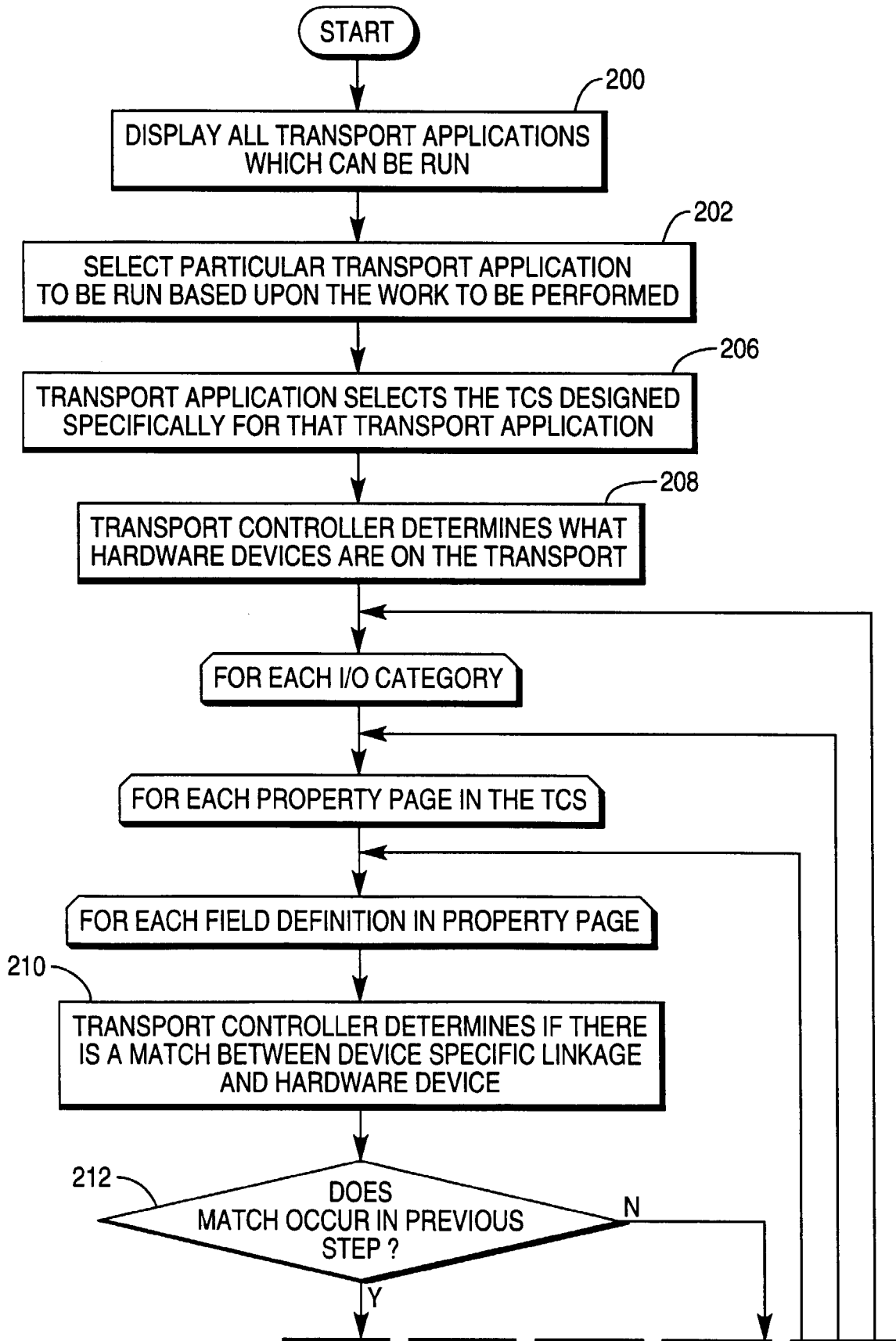
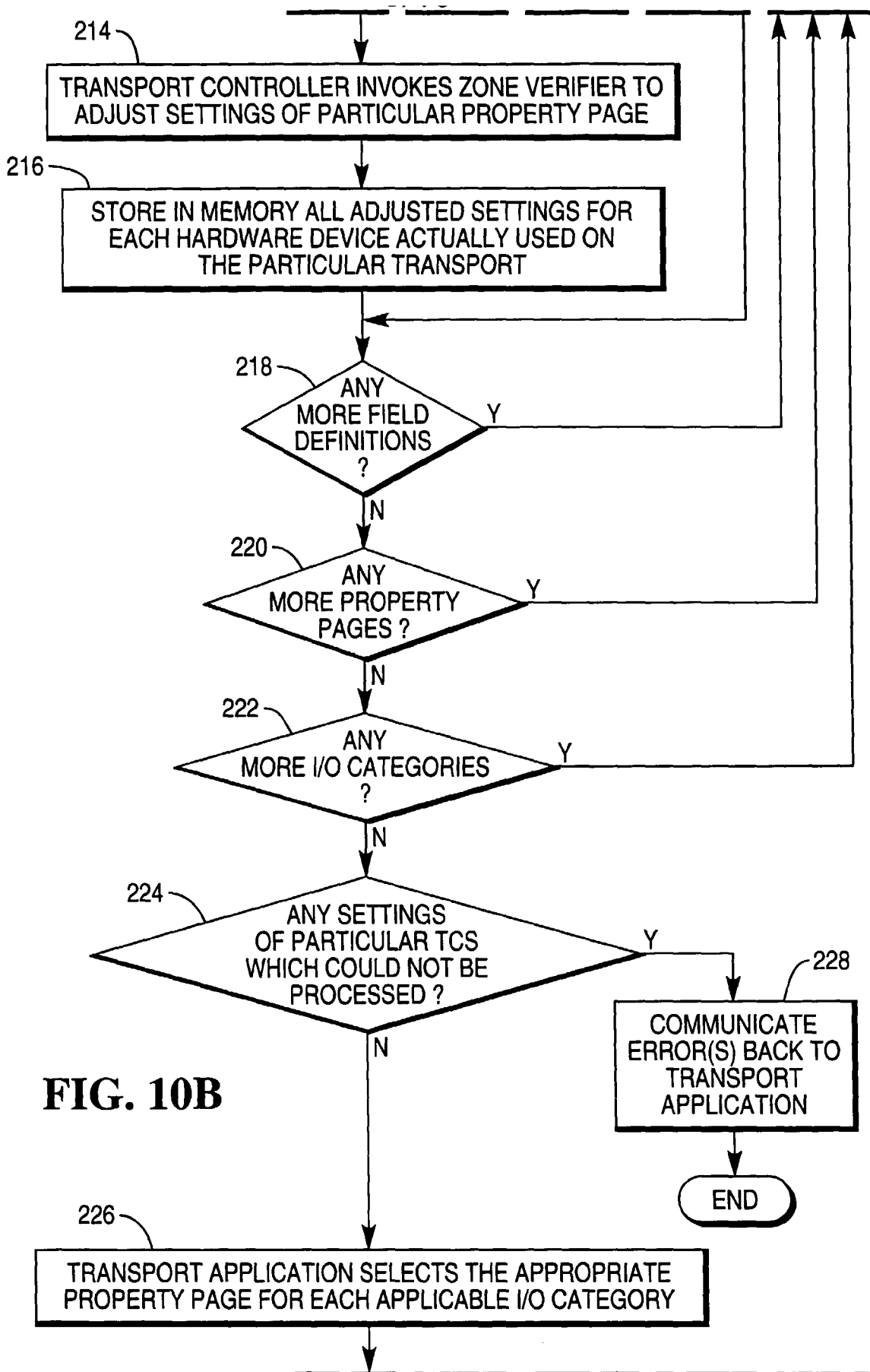
FIG. 9

FIG. 10A

**FIG. 10B**

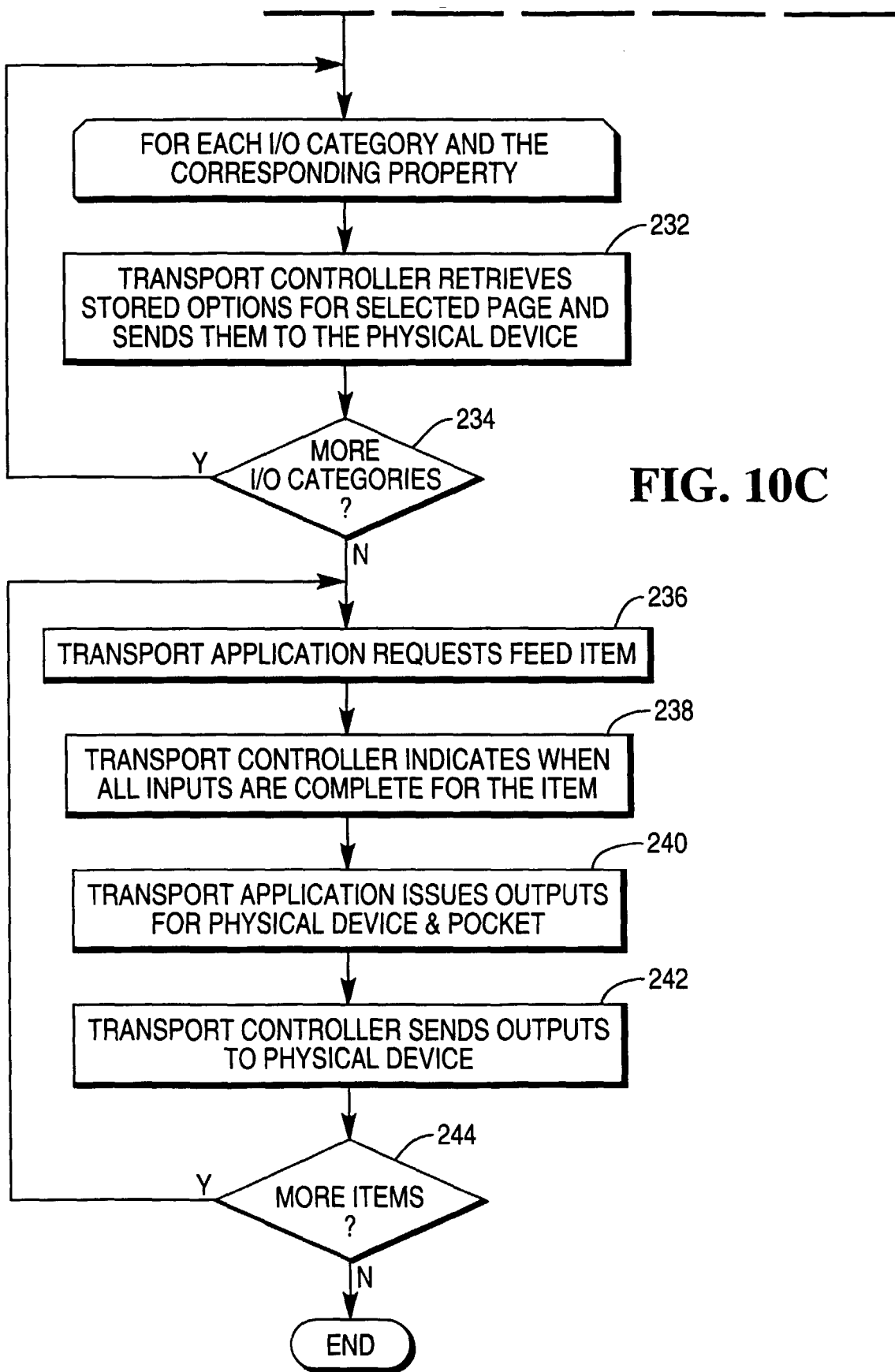
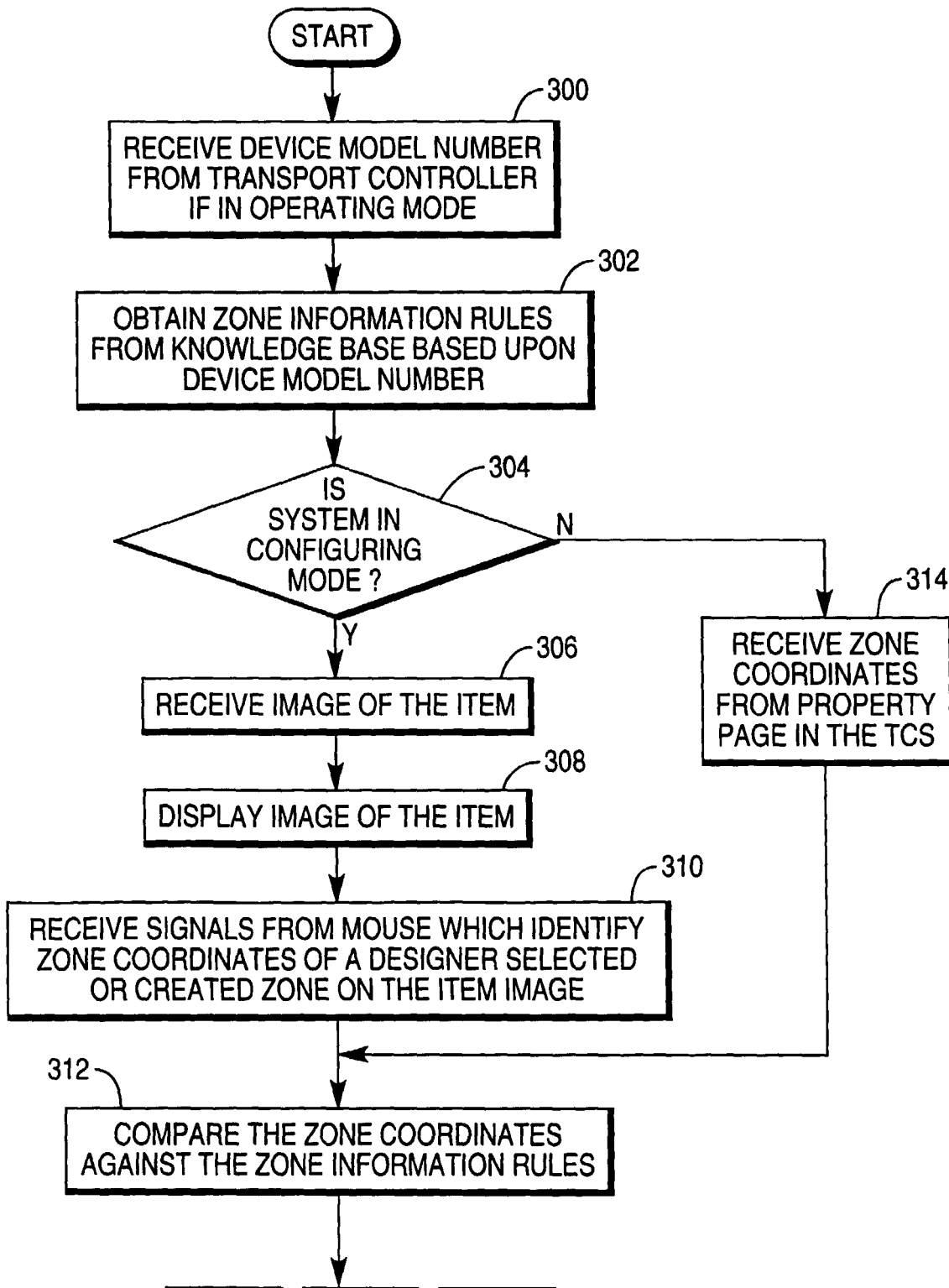


FIG. 11A



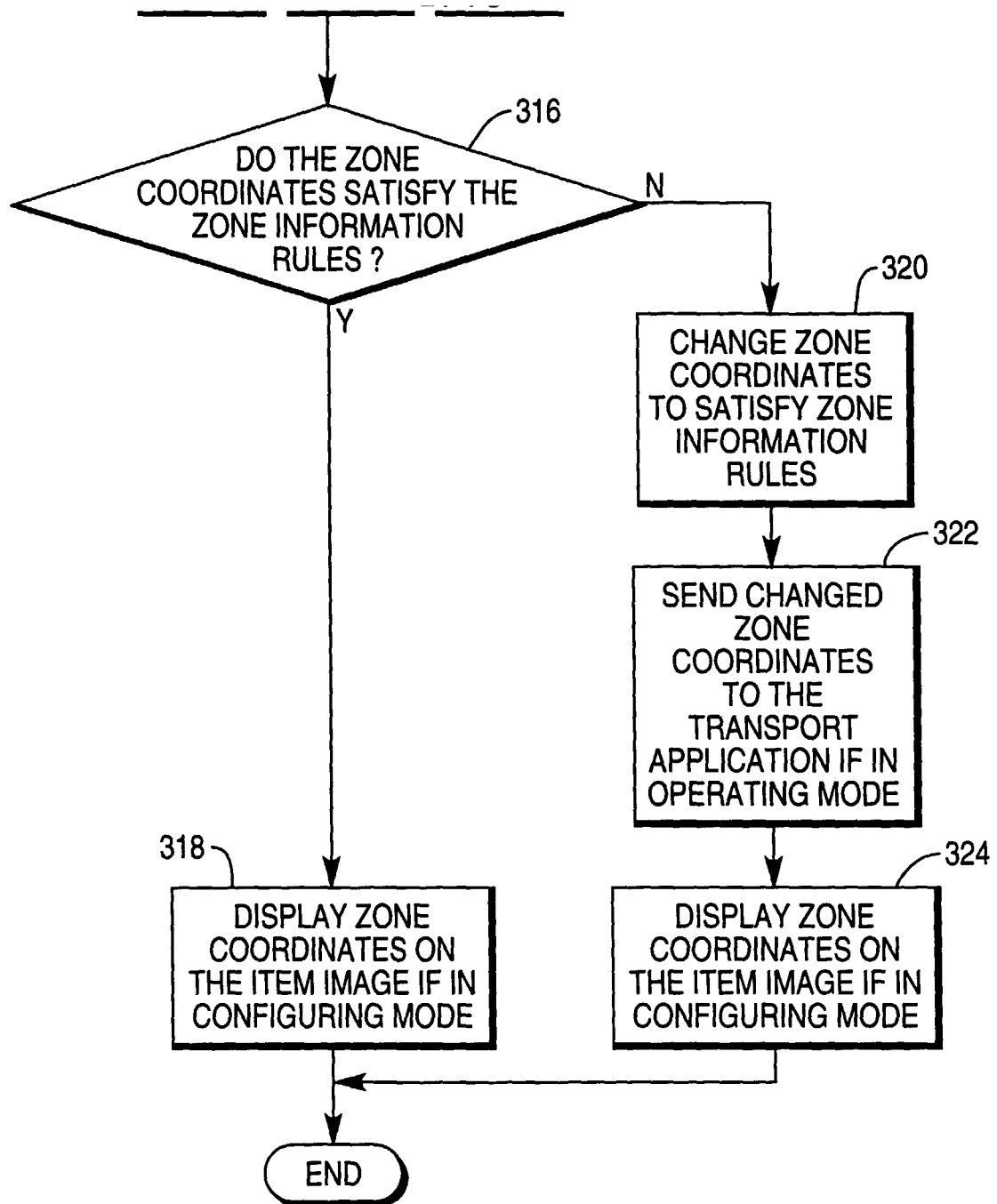
**FIG. 11B**

FIG. 12A

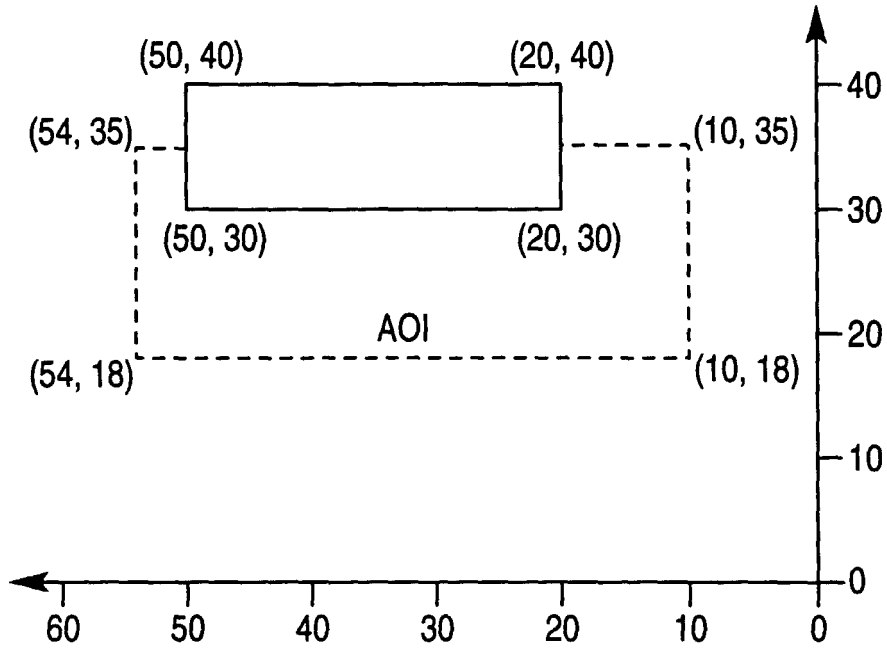


FIG. 12B

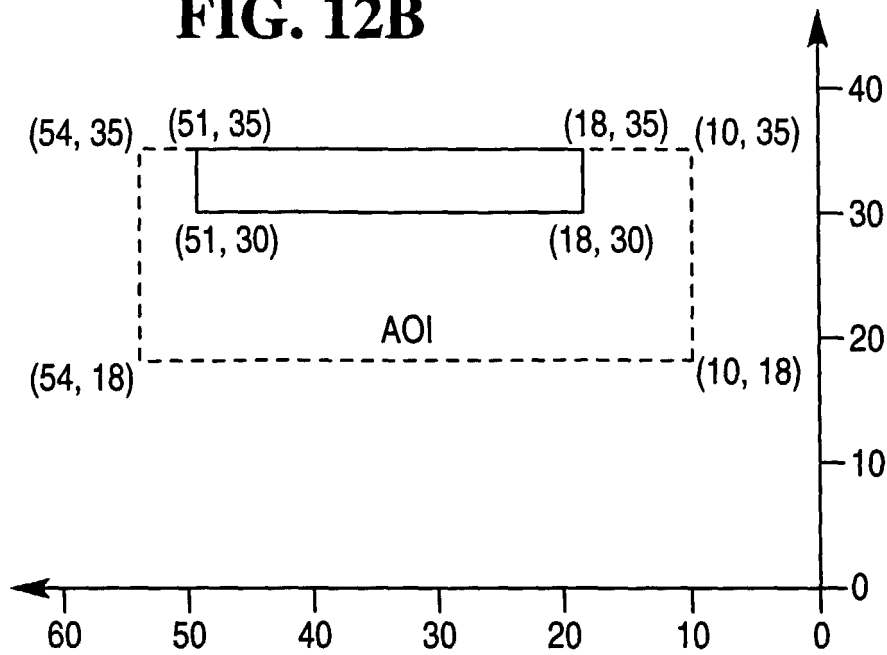


FIG. 13A

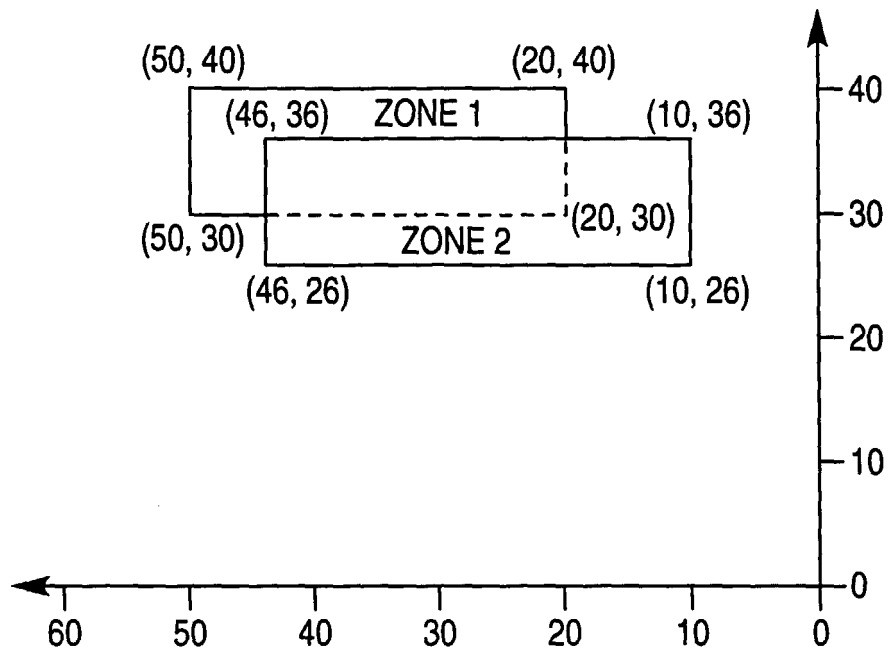


FIG. 13B

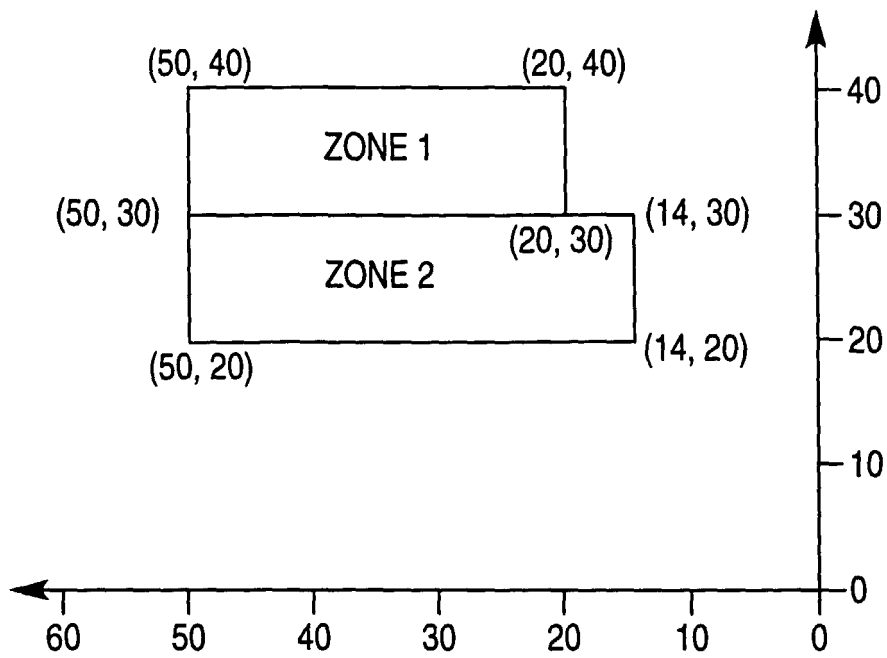


FIG. 14A

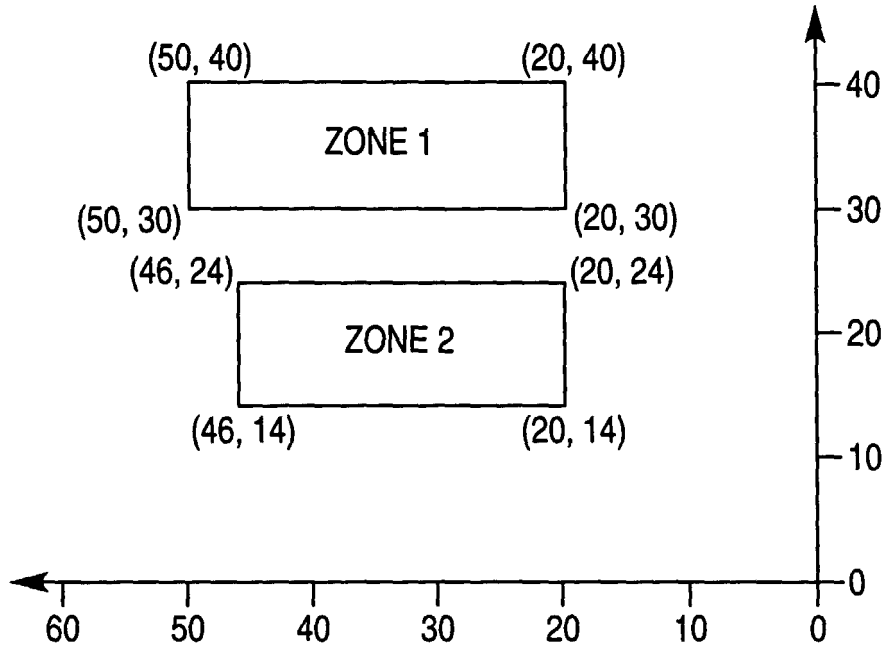


FIG. 14B

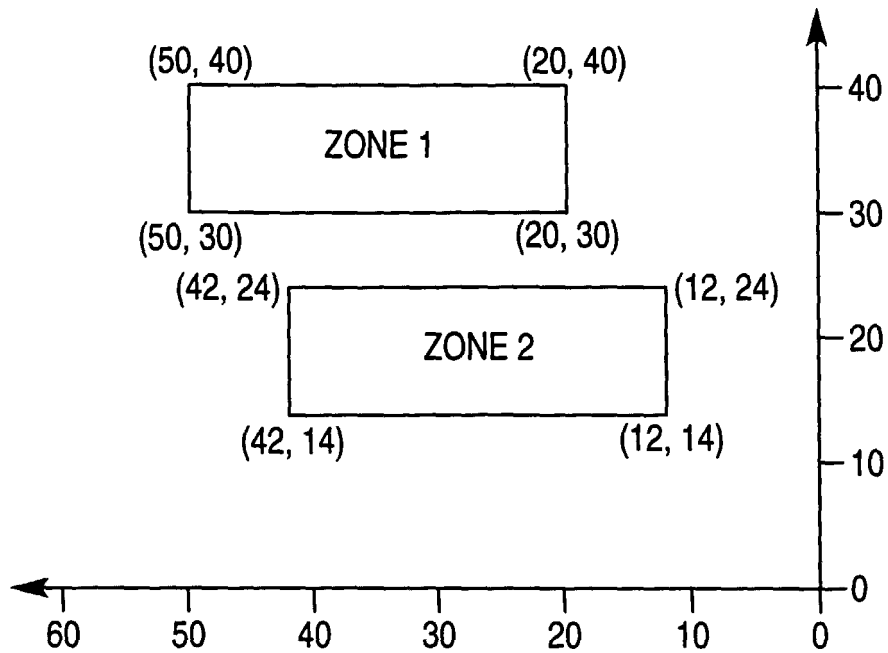


FIG. 15A

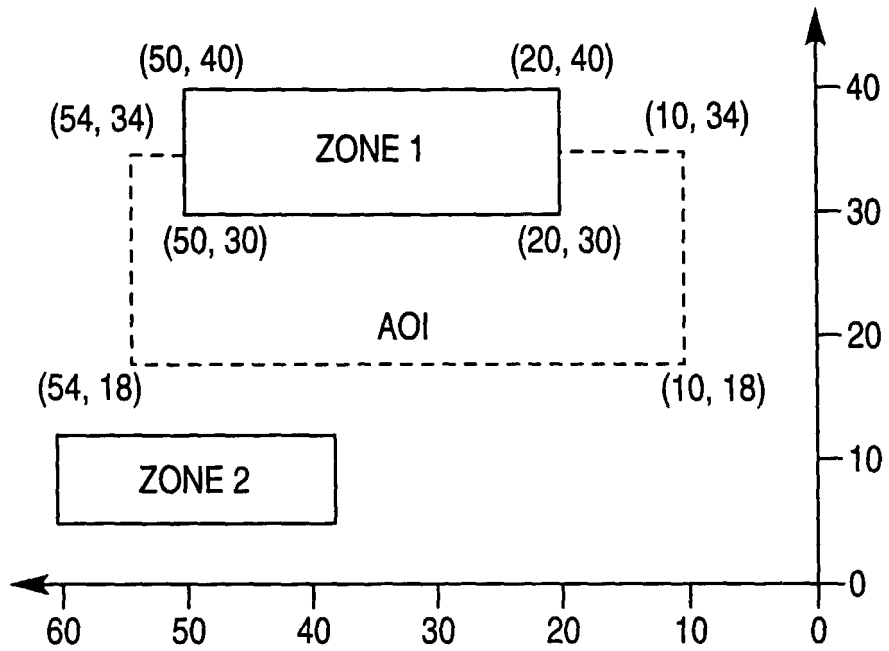


FIG. 15B

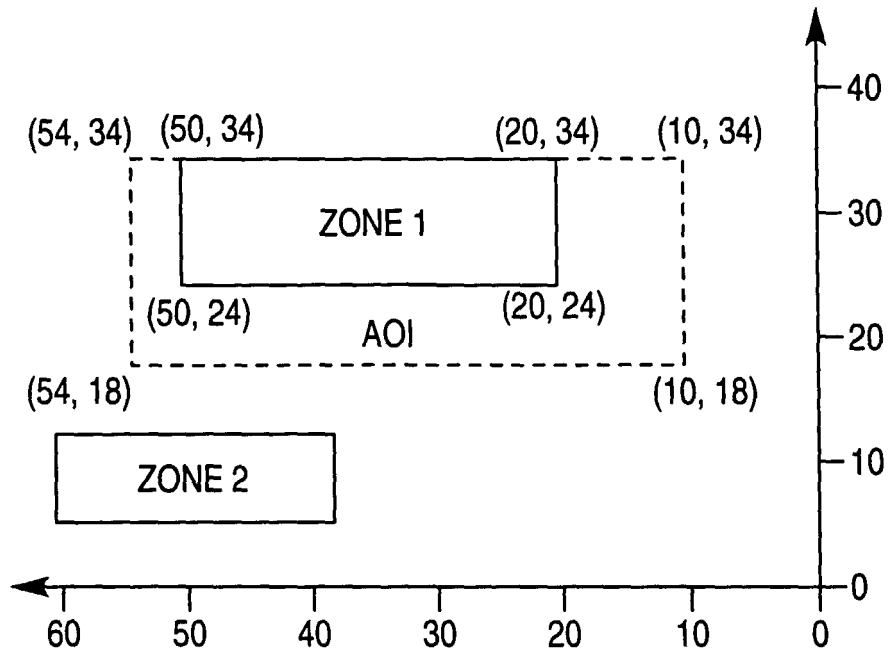


FIG. 16A

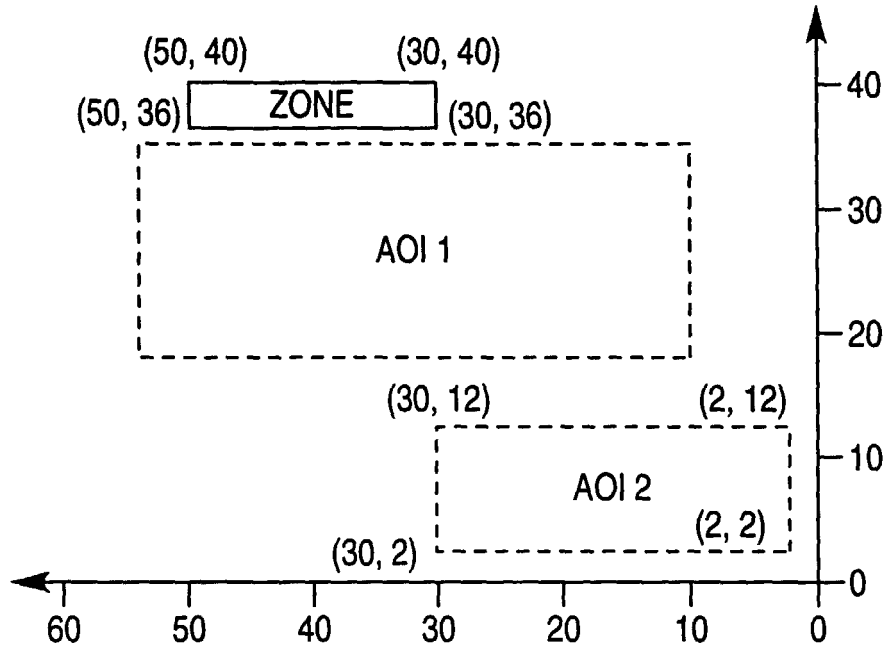


FIG. 16B

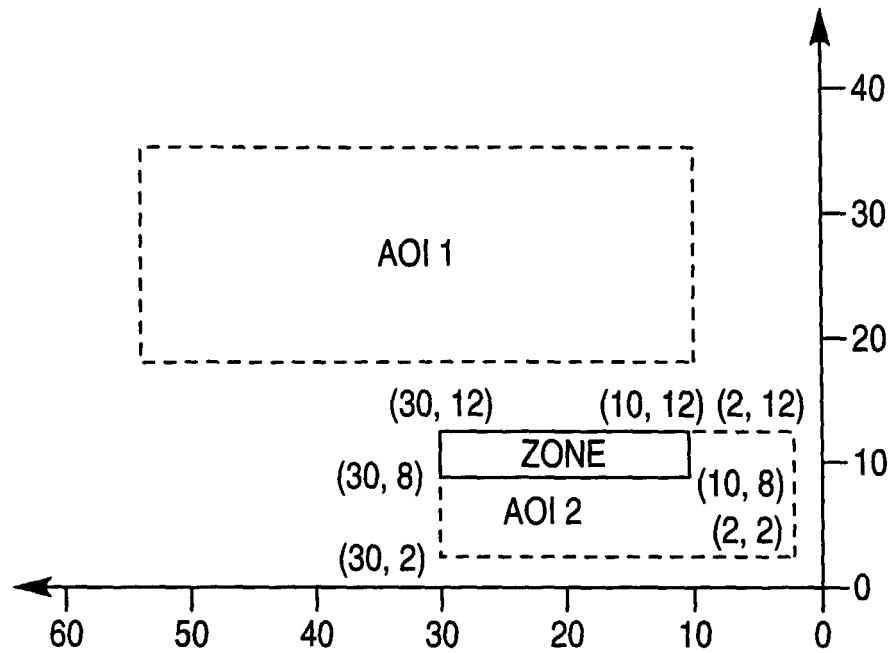


FIG. 17

